Pengantar Library, Classpath, dan Project

Sebagai patokan, versi NetBeans yang digunakan Penulis adalah versi 7.0, tetapi pada prinsipnya keluarga versi 7.x juga bisa digunakan, bahkan versi sebelumnya pun bisa, hanya menyesuaikan beberapa hal teknis saja, misal nama *library*.

Hal yang paling mendasar dan harus dikuasai oleh Programmer Java adalah memasang *library* (kelas dalam bentuk biner). Karena Java yang dirancang secara *object oriented,* tentunya program yang terdiri atas kesatuan antar-objek(satu atau lebih) yang membentuk aplikasi. Sehingga sebuah objek tertentu bisa digunakan dalam beberapa aplikasi yang berbeda. Objek ini biasanya bersifat umum penggunaannya, misalkan objek kalender, objek driver database, objek report, dan lain-lain.

1.1 Pengaturan ClashPath

Path adalah pemetaan suatu direktori pada sistem operasi agar direktori tersebut bisa dikenal secara global pada sistem. Sedangkan ***Classpath*** untuk memberitahukan pada Java bahwa sebuah *class* java tertentu (bisa berbentuk .jar atau .*class*) kita simpan pada direktori/alamat file tertentu.

Contoh classpath: Ada sebuah file jar yang bisa dipakai untuk driver database MySQL dengan alamat filenya adalah **c:\libku\mysql-con-bin.jar** maka untuk mengenalkan pada mesin Java tentang driver tersebut adalah dengan melakukan setting ***ClassPath***.

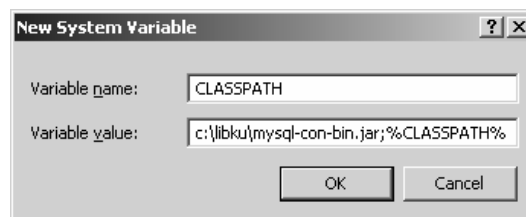
1.1.1 Pengaturan ClassPath dari Dos Prompt

Untuk melakukan pengaturan **ClassPath**, Anda dapat membuka jendela DOS Prompt, perintahnya adalah:

```
SET CLASSPATH = c:\libku\mysql-con-bin.jar;%CLASSPATH%
```

1.1.2 Pengaturan ClassPath dari Windows XP

Untuk melakukan pengaturan *ClassPath* dari Windows XP, Anda bisa mengikuti langkah berikut:

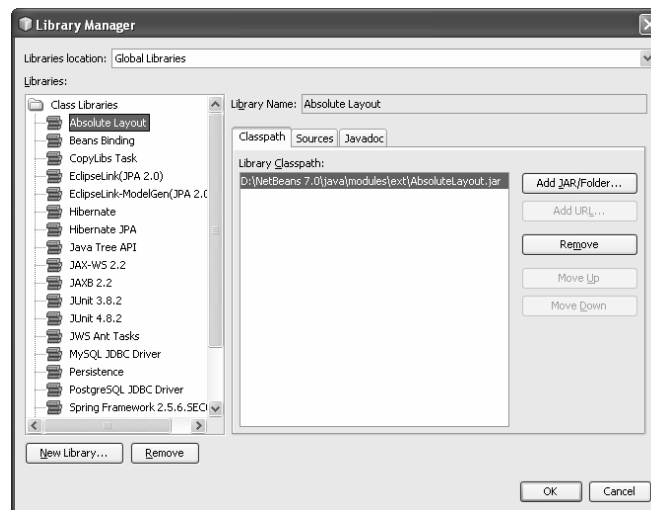
1. Jalankan dari menu **Control Panel → System**, sehingga muncul jendela **System Properties**.
2. Pilih tab **Advanced**, lalu pilih tombol **Environment Variables**.
3. Pada **system variables**, tekan tombol **New**.
4. Pada jendela **New System Variable**, masukkan nama variable pada isian **Variable name**, kemudian isi variabelnya pada isian **Variable value**.

Gambar 1.1 Setting classpath pada Windows XP

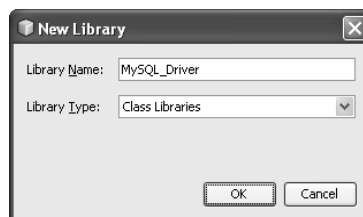
1.2 Library

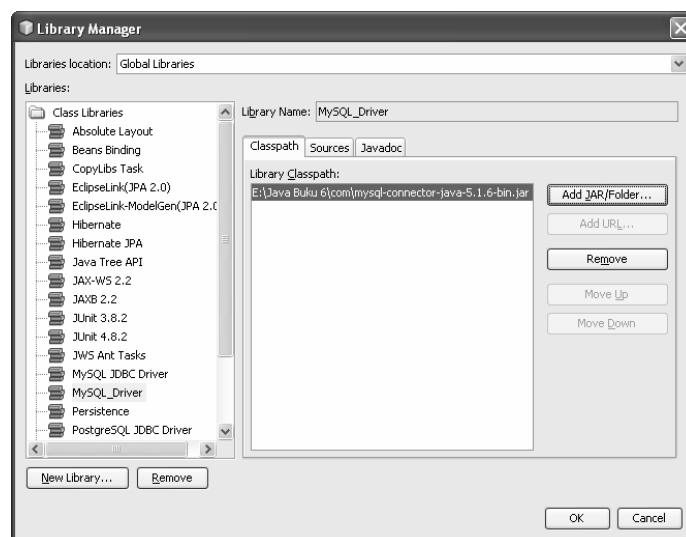
Pemasangan ***ClassPath*** melalui **Command Prompt** dan Windows, sudah jarang digunakan, kecuali keperluan tertentu. Misalnya membuat shortcut atau membuat installer aplikasi Java yang akan berjalan pada komputer pelanggan dan lain sebagainya. Yang paling sering dialami oleh programmer adalah memasang *classpath* tersebut dalam editornya (dalam hal ini NetBeans), yang lebih dikenal dengan istilah ***Library***. Secara umum cara pemasangan *library* adalah sama, yakni menyiapkan filenya lalu memasangnya.

1.2.1 Membuat Library

Sebuah ***Library*** bisa terdiri atas beberapa kelas. Dalam hal ini, contoh *Library* yang dibuat adalah driver MySQL Connector (untuk koneksi dengan database MySQL) yang diperoleh dari internet, bukan dengan driver yang tersedia dalam NetBeans. Ikuti langkah-langkah berikut:

1. Copy file **mysql-connector-java-5.1.6-bin.jar** dari <**CD Buku**>**\com**, kemudian letakkan dalam komputer Anda.
2. Masuk jendela **Library Manager**, melalui menu **Tool → Libraries.**

Gambar 1.2 Jendela Library Manager

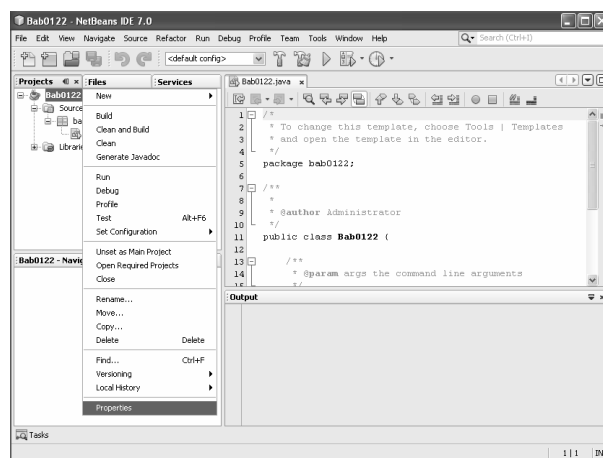
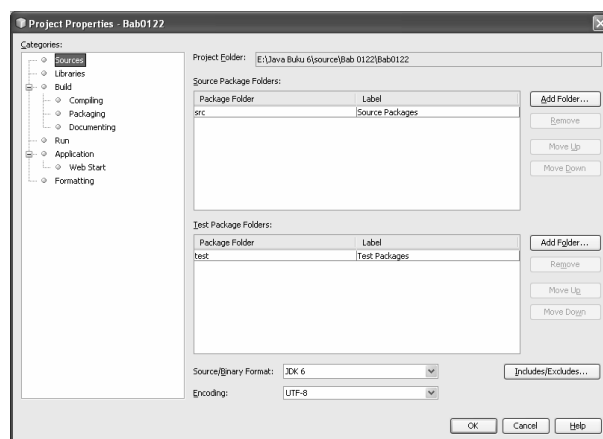
3. Dalam jendela **Library Manager**, klik tombol **New Library...**, sehingga muncul jendela **New Library**.

Gambar 1.3 Jendela Library Manager

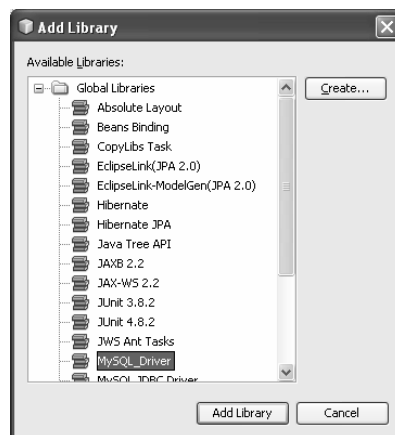
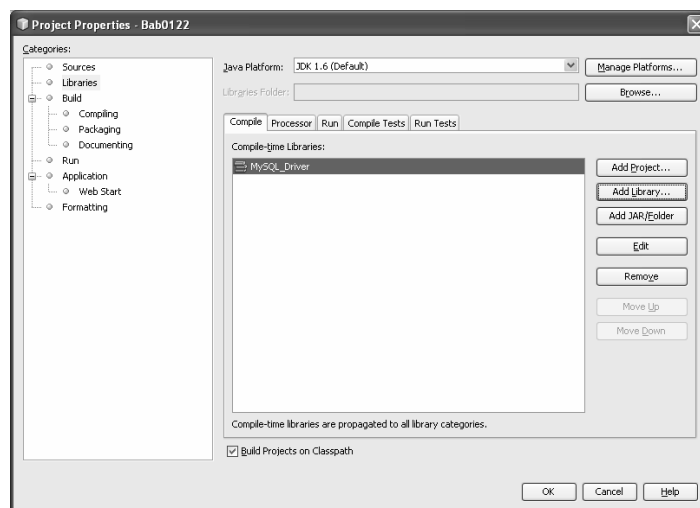
4. Dalam jendela **New Library**, masukkan **MySQL_Driver** dalam kotak masukan **Library Name**, kemudian tekan **OK**. Sehingga nama librari **MySQL_Driver** masuk dalam daftar **Class Libraries** pada jendela **Library Manager**.
5. Setelah nama librari **MySQL_Driver** terpasang, langkah selanjutnya adalah mengisi kelas-kelas yang akan dimasukkan dalam *library* ini. Masih dalam jendela **Library Manager** dan **MySQL_Driver** terpilih. Perhatikan jendela sebelah kanan, pada tab **ClassPath**, klik **Add JAR/Folder**, sehingga muncul jendela **Browse JAR/Folder**. Pilih file **mysql-connector-java-5.1.6-bin.jar**, klik **Add JAR/Folder**. Sehingga kelas tersebut masuk dalam librari yang sedang dibuat. Akhiri dengan mengklik tombol **OK**.

Gambar 1.4 Jendela Library Manager Paska Pemasangan Kelas

1.2.2 Memasang Library pada Project

Setelah sebuah library dibuat, library tersebut belum berarti apa pun tanpa adanya **Project** yang menggunakannya. Sehingga setiap Project perlu dihubungkan dengan library tersebut jika ingin memanfaatkan kelas-kelas yang disimpannya. Untuk memasang library dalam sebuah **Project**, lakukan langkah-langkah berikut:

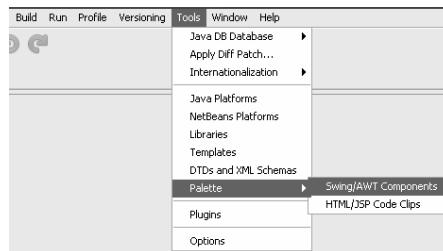
1. Buka **Project** Anda. Kalau belum punya, buatlah **Project** baru, **File→ New Project**, sehingga muncul jendela **New Project**. Dalam kotak **Categories** pilih **Java**, dalam kotak **Projects** pilih **Java Application**. Klik tombol **Next**, sehingga muncul jendela **New Java Application**. Isikan **Bab0122** dalam kotak isian **Project Name**, klik **Finish**.

2. Klik kanan **Project→Properties**, sehingga muncul jendela **Project Properties**.

Gambar 1.5 Menu Project Properties

Gambar 1.6 Project Properties Awal

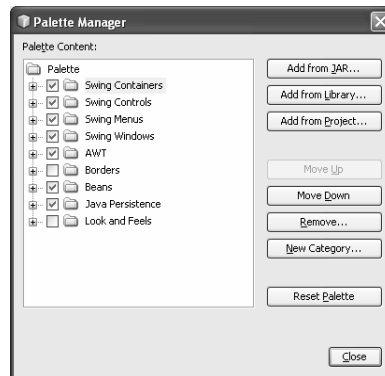
3. Dalam jendela **Project Properties**, pada kotak **Categories** pilih **Libraries**. Pada sebelah kanan tekan tombol **Add Library...** (pada tab **Compile**) sehingga muncul jendela **Add Library**. Pilih **MySQL_Driver**, tekan **Add Library**, kemudian tekan **OK** pada jendela **Properties**. Untuk NetBeans versi 6 ke atas, Anda bisa menggunakan driver MySQL yang sudah disertakan editor tersebut (**MySQL JDBC Driver**).

Gambar 1.7 Memilih Library

Gambar 1.8 Library Terpasang pada Project Properties

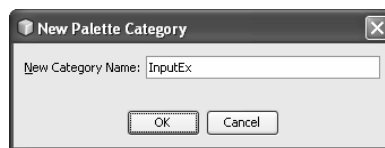
1.2.3 Memasang Library pada Palette

Meskipun Anda sudah memasang jar file dalam library, jika kelas tersebut bertipe visual (*control design*), maka Anda perlu memasangnya dalam **Palette** agar bisa dipakai saat design form. Seperti contoh kelas **JCalendar** (menampilkan pilihan tanggal), JTextEx, JPasswordEx dan ComboUI (JCombobox yang bisa terhubung ke database). Lakukan langkah-langkah berikut ini:

1. Copy contoh file-file berikut (.jar) dalam folder <**CD Buku**>**\com,** ke dalam komputer; **jcalendar-1.3.2.jar** dan **DBPaket-6.0.jar**.
2. Dari menu utama NetBeans pilih menu **Tools** → **Palette** → **Swing/AWT Components,** sehingga muncul **Palette Manager.**

Gambar 1.9 Menu Palette

Akan ditampilkan jendela **Palette Manager**. Dalam jendela tersebut terdapat daftar **Palette** berdasarkan kategori kelas masing-masing, antara lain:

- **Swing Containers** untuk meletakkan komponen lain, seperti sebuah papan.
- **Swing Controls** adalah objek kendali, seperti JButton, Jcombobox, JRadioButton, JLabel, JCheckBox, JTextField, JList, JTextArea, dan lain-lain.
- **Swing Menus** untuk kelas-kelas menu, seperti JmenuBar, JMenuItem, dan lain-lain.
- **Swing Windows** untuk kelas-kelas jendela, seperti JFrame, JDialog, JFileChooser, dan lain-lain.
- **AWT** untuk kelas-kelas AWT (sebelum versi swing).
- **Borders**, **Beans**, **Java Persistence**, dan **Look** and **feels**.

Gambar 1.10 Jendela Palette Manager

3. Tambahkan kategori baru dengan cara klik tombol **New Category...** sehingga muncul jendela **New Palette Category**. Masukkan nama kategorinya **InputEx**, lalu klik tombol **OK**.

Gambar 1.11 Jendela New Palette Category

4. Setelah kategori sudah jadi, pada jendela **Palette Manager**, pilih palet yang baru dibuat (InputEx), kemudian klik tombol **Add From JAR**. Cari dan pilih **DBPaket-6.0.jar** sebelumnya, klik tombol **Next**, sehingga sampai pada langkah kedua memilih komponen (**2. Select Components**). Pada daftar **Available Components** pilih **ComboUI, JPasswordEx, JTextExt** lalu klik tombol **Next**.

   Apabila Anda tidak menemukan kelas yang bisa dipilih, berarti hasil kompilasi atau file jar-nya belum menghasilkan objek kelas. Silakan cek dengan baik *source* dari kelasnya, kemudian **Build and clean** ulang.

5. Pada langkah **3. Select Palette Category**, pilih kategori **InputEx,** lalu klik tombol **Finish.** Sekarang Anda mendapati jendela **Palette Manager** lagi yang sudah dilangkapi dengan komponen baru, klik tombol **Close** untuk menutup jendelanya.

Anda bisa meletakkan dalam kategori apa saja pada palet yang sudah ada, tetapi lebih baiknya semua kelas berada dalam paket yang sesuai agar pembuatan aplikasi berikutnya tidak mengalami kesulitan. Dan tentunya lebih tertata dengan baik seperti contoh yang dilakukan oleh **Sun** dalam **NetBeans**.

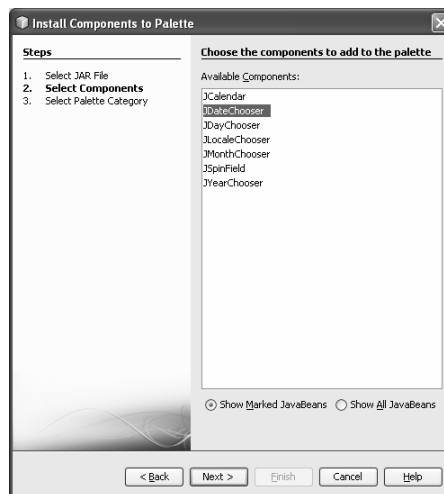
Gambar 1.12 Langkah Kedua

Gambar 1.13 Langkah Terakhir

6. Selanjutnya Anda bisa menggunakan komponen tersebut seperti halnya komponen lainnya dalam **Palette NetBeans**.

Gambar 1.14 Komponen Baru

Komponen baru akan muncul dalam jendela **Palette Manager**.

7. Selanjutnya lakukan hal yang sama pada paket **jcalendar-1.3.2.jar** agar dalam palet Anda terdapat pilihan sebuah *date picker* (objek untuk memilih tanggal). Pada langkah kedua, pilih satu komponen saja, yaitu **JDateChooser.** Selanjutnya klik **Next,** dan pilih kategori seperti sebelumnya.

Gambar 1.15 Memilih Komponen Tanggal

Gambar 1.16 Komponen Baru dalam Paket InputEx

1.3 Memastikan Library Bekerja dalam Proyek

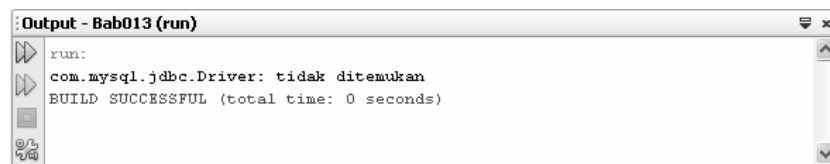
Meskipun **Project** Anda secara kompilasi hasilnya tidak ada kesalahan (sintak error), namun tidak menutup kemungkinan terjadi kesalahan saat dijalankan (*run time error*). Salah satunya adalah ***Library*** yang terpasang tidak berjalan dengan baik, atau ada kesalahan dalam memilih file Jar yang dimaksud, atau adanya perbedaan versi dengan peranti modul lainnya, dan sebagainya.

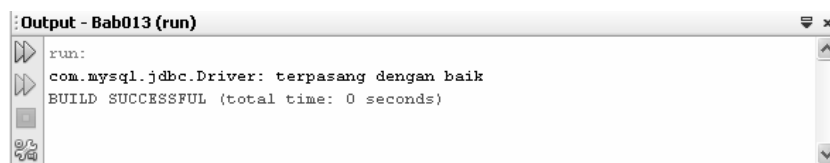
Cara yang paling mudah untuk memastikan Library berjalan dengan baik atau tidak adalah dengan menguji kelas yang bersangkutan ke dalam program. Berikut ini untuk menguji kelas **Driver Connection**, ikuti langkah-langkahnya:

1. Buat **Project** baru, **File→ New Project**, sehingga muncul jendela **New Project**. Dalam kotak **Categories** pilih Java, dalam kotak **Projects** pilih **Java Application**. Klik tombol **Next**, sehingga muncul jendela **New Java Application**, isikan **Bab013** dalam kotak isian **Project Name**, klik **Finish**.
2. Tambahkan kode dalam kelas **Bab013**, seperti berikut:

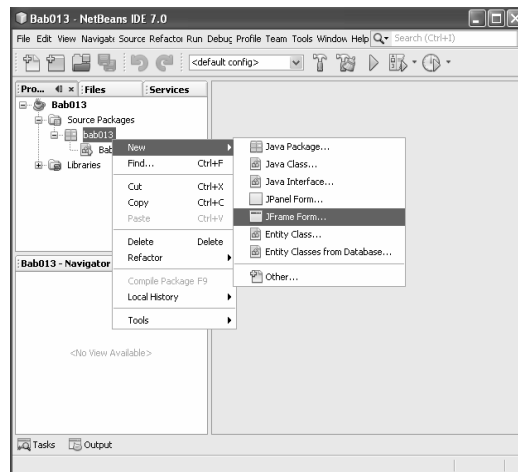
```
package bab013;

public class Bab013{
    /**
     * @param args the command line arguments
     */
```

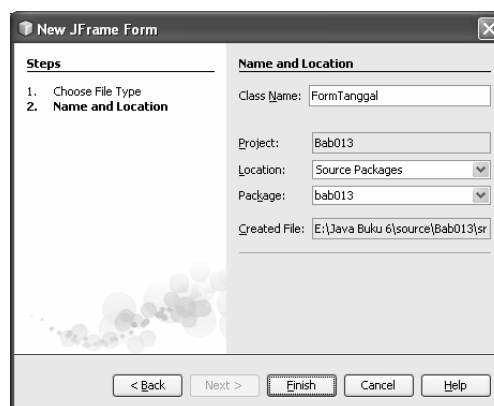
```
    public static void main(String[] args) {
        // TODO code application logic here
       String mKelas="com.mysql.jdbc.Driver";
        try {
            Class.forName(mKelas);

            System.out.println(
                   mKelas + ": terpasang dengan baik");
       }

       catch(ClassNotFoundException e) {
            System.out.println(
               mKelas + ": tidak ditemukan");
       }
    }
}
```

3. Lakukan *Clean And Build*, kemudian jalankan. Jika ada pesan dalam jendela *Output* muncul seperti berikut.

Gambar 1.17 Jendela Output Pertama

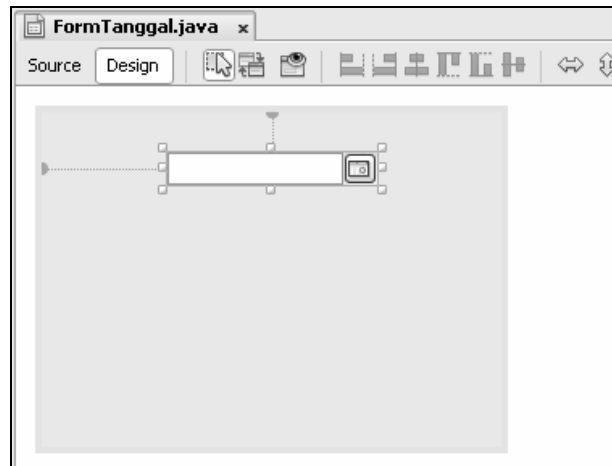
Munculnya pesan "tidak ditemukan" berarti ***library*** belum terpasang dengan baik. Solusinya, harus dicek melalui jendela **Project Properties**. Pada kategori **Libraries** seperti pada pembahasan sebelumnya, jika belum ada library yang terpasang dengan baik, berarti harus dipasang dulu, atau dicek isi librarinya. Lakukan kompilasi lagi, jalankan, sehingga jendela Output menampilkan pesan seperti berikut.

Gambar 1.18 Jendela Output Kedua

Untuk melakukan pengujian pada kelas **JCalendar**, ikuti langkah-langkah seperti berikut:

1. Masih dalam project **Bab013**, buat form baru dengan cara: klik kanan paket **bab013 → New → JFrame Form…**

Gambar 1.19 Menu Menambah Form

2. Dalam jendela **New JFrame Form**, masukkan nama kelasnya **FormTanggal** pada kotak isian **Class Name**. Klik **Finish**.

Gambar 1.20 Jendela JFrame Form

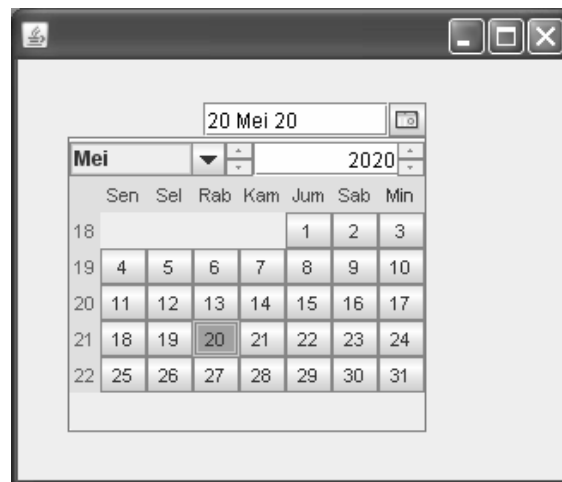
3. Sebelum melakukan design form, buka jendela **Palette**. Jika komponen **JCalendar** muncul, berarti pemasangan ***Library*** berhasil dengan baik. Ambil sebuah objek **JDateChooser**,

dengan cara klik kemudian arahkan kursor ke dalam form, lalu lepaskan dalam area form.

Gambar 1.21 Jendela JFrame Form

4. Edit kode program pada kelas **Bab013.java**, menjadi seperti berikut ini.

```
package bab013;

/**
 *
 * @author Administrator
 */
public class Bab013 {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
       try {
            Class.forName("com.mysql.jdbc.Driver");

            System.out.println(
                   "driver terpasang dengan baik");
       }

       catch(ClassNotFoundException e) {
            System.out.println(
               "jdbc.Driver tidak ditemukan");
       }
```

```
        FormTanggal frm = new FormTanggal();
        frm.setVisible(true);
    }
}
```

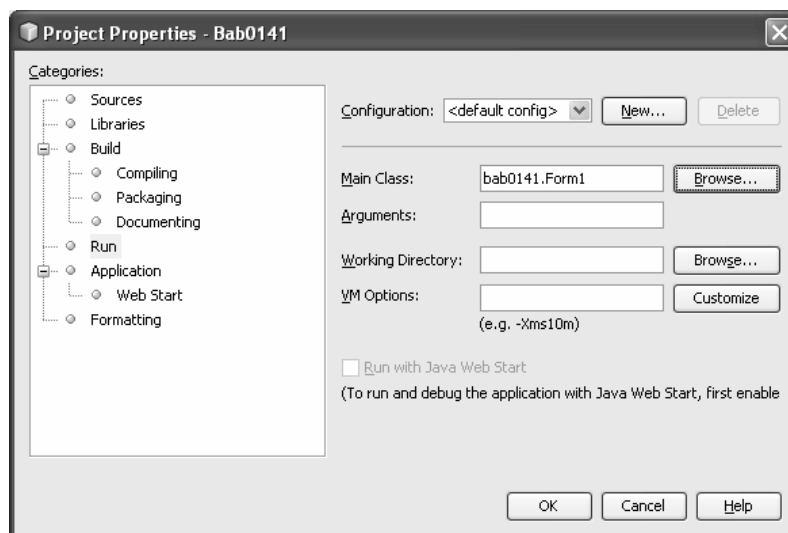
5. Lakukan **Compile And Build**, jalankan program. Apabila objek **JCalendar** tersebut berjalan dengan baik, berarti Anda sudah bisa memasang ***Library*** dengan benar.

Gambar 1.22 Tampilan Objek JCalendar

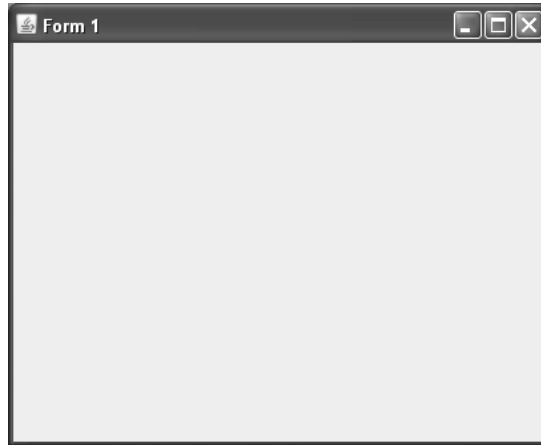
1.4 Mengelola Project

Project adalah hal yang cukup penting dalam NetBeans. Semua aplikasi yang bisa di-debug, disimpan dalam Project. Informasi dalam Project ini akan digunakan untuk menentukan hasil akhir dari kompilasi yang dihasilkan saat *Clean and Build*.

1.4.1 Startup Project

Project terdiri atas beberapa kelas di dalamnya. Setiap kelas boleh memiliki metode main sehingga secara teori, bisa lebih dari satu kelas yang mampu berdiri sendiri (dengan metode main masing-masing). Sehingga sebuah Project perlu untuk ditentukan, mana kelas yang akan dijalankan pertama kali. Ikuti langkah-langkah berikut.

1. Buat project baru (Java Application), beri nama **Bab0141**.
2. Tambahkan sebuah **JFrame Form**, ke dalam paket **bab0141**, beri nama **Form1**.
3. Lakukan lagi pada langkah nomor dua agar Project tersebut terdiri atas beberapa kelas (dengan nama yang berbeda).
4. Ubah properti **Title** pada masing-masing form agar terlihat perbedaannya saat dijalankan.
5. Masuk dalam jendela **Properties**, melalui jendela **Project → Properties**, sehingga muncul jendela **Properties**.

Gambar 1.23 Memilih Kelas Main

6. Pada jendela **Project Properties**, pilih **Run** pada kotak pilihan **Categories**. Tentukan kelas main dengan cara mengklik tombol **Browse....** Klik **OK**.
7. Jalankan program dan lihat hasilnya.

Gambar 1.24 Form1 Sebagai Main Class

Gambar 1.25 Form2 Sebagai Main Class

1.4.2 Direktori Kerja Project

Direktori kerja sebuah aplikasi menentukan letak direktori yang dikenali oleh aplikasi tersebut secara *default*. Nah, direktori tersebut bisa disetting melalui **Project Properties**. Ada perbedaan antara direktori kerja aplikasi saat berjalan (melalui file .jar) dan aplikasi berjalan melalui NetBeans. Ikuti langkah-langkah berikut:

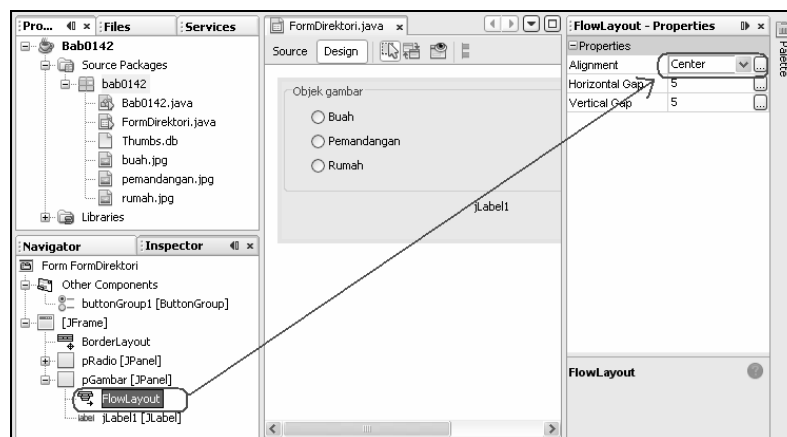
1. Buat project baru (Java Application), beri nama **Bab0142**.
2. Tambahkan sebuah **JFrame Form**, ke dalam paket **bab0142**, beri nama **FormDirektori**.
3. Tambahkan beberapa objek ke dalam form sehingga menjadi seperti berikut.

Gambar 1.26 Design Form Direktori

Berikut ini data objek dan propertinya.

Object [Variable Name]	**Property**	**Value**
JFrame	title	Direktori kerja
	SetLayout →BorderLayout	
JPanel[pRadio]	border/Title	Objek gambar
	Layout-Direcction	First
	Accessible Parent	Form
ButtonGroup [buttonGroup1]		
JRadioButton[r_buah]	text	Buah
	Accessible Parent	pRadio
	buttonGroup	buttonGroup1
JRadioButton [r_pemandangan]	text	Pemandangan
	Accessible Parent	pRadio
	buttonGroup	buttonGroup1
JRadioButton[r_rumah]	text	Rumah
	Accessible Parent	pRadio

	buttonGroup	buttonGroup1
JPanel[pGambar]	Layout-Direcction	Center
	Accessible Parent	Form
	Set Layout→Flow Layout (Alignment=center)	
JLabel [jLabel1]	text	jLabel1
	Layout-Direcction	Center
	Accessible Parent	pGambar
	preferredSize	[300, 400]

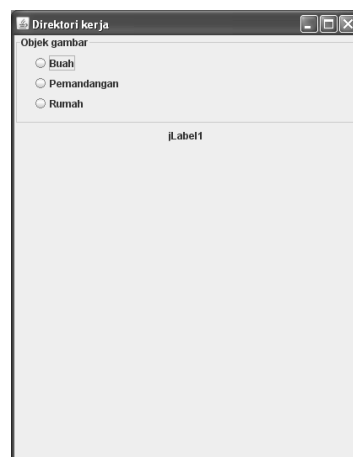
Tanda abu-abu pada tabel menunjukkan setting pada objek yang bersangkutan, dengan cara klik kanan lalu ikuti perintahnya. Untuk objek JFrame berarti klik kanan pada ruang kosong objek form → **Set Layout → Border Layout**. Untuk keterangan dalam kurung (seperti Alignment=center) adalah Property dari Layout-nya. Untuk mengubahnya, pilih objeknya kemudian menuju jendela **Inspector**. Pilih layoutnya (Flow layout), selanjutnya lihat jendela **Properties**.

Gambar 1.27 Property Layout

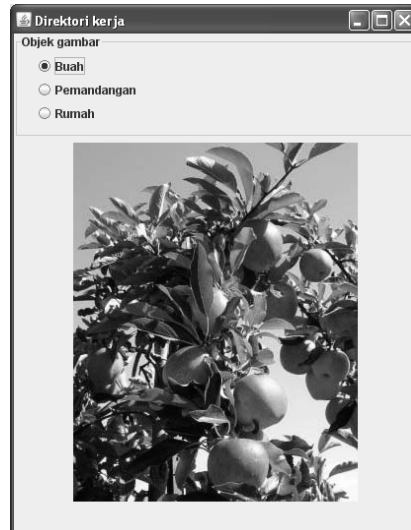
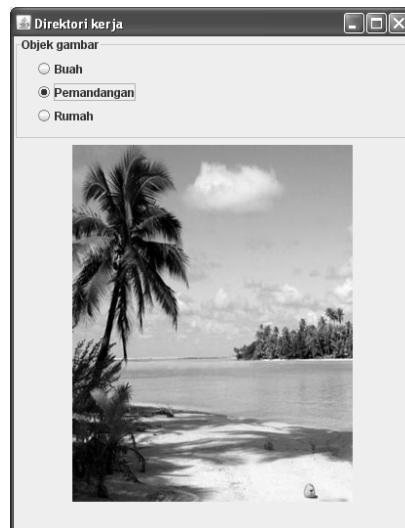
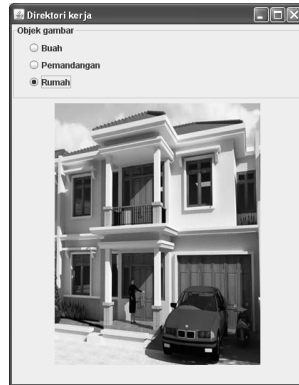
4. Tambahkan metode pada objek **r_buah → Events → Item → itemStateChanged**. Berikut kode programnya:

```
private void r_buahItemStateChanged
                     (java.awt.event.ItemEvent evt) {

    jLabel1.setText("");
    String strFile="";

    if (r_buah.isSelected()){
       strFile="./src/bab0142/buah.jpg";
    }
    else
    if (r_pemandangan.isSelected()){
       strFile="./src/bab0142/pemandangan.jpg";
    }
    else
    {
       strFile="./src/bab0142/rumah.jpg";
    }

    jLabel1.setIcon(new ImageIcon(strFile) );
 }
```

Folder tidak disebutkan secara lengkap agar bisa dinamis, yakni cukup mengacu pada sebuah folder atau *working directory*.

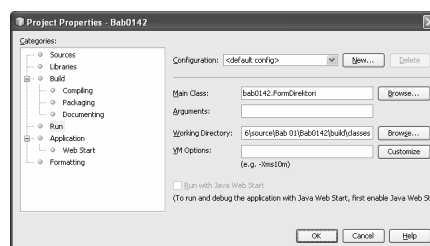
5. Copy file **buah.jpg**, **pemandangan.jpg** dan **rumah.jpg** dari <CD buku>\source\Bab 01\bab0142\src\bab0142, ke dalam paket **bab0142** (contoh peletakannya seperti *source code* dalam CD buku Anda).

6. Lakukan *Clean and Build*, dan jalankan program.

Gambar 1.28 Tampilan Awal

Pada saat pertama kali muncul, form ini tidak menampilkan gambar karena belum ada objek gambar yang terpilih. Untuk itu pilih salah satu objek dan perhatikan gambar yang muncul.

Gambar 1.29 Memilih Gambar Buah

Gambar 1.30 Memilih Gambar Pemandangan

Gambar 1.31 Memilih Gambar Rumah

7. Jika Pembaca benar dalam setiap langkahnya, akan mendapatkan bahwa gambar yang muncul dalam form berganti-ganti sesuai dengan pilihan nama objek gambar.

   Dalam hal ini, **NetBeans** menganggap direktori utama (*home*) adalah dalam folder project (jadi satu dengan file **build.xml**), lebih tepatnya di atas (*parent*) folder **src**. Akan tetapi, ada hal yang harus dipahami, jika Anda bertujuan untuk mempublikasikan aplikasi ini, maka perlu dirancang peletakan folder di luar **src**.

8. Anda juga bisa memindahkan direktori kerja project ini ke dalam folder yang lain, misal **./classes**. Kemudian pindahkan berkas-berkas gambar ke dalam forlder ini. Lakukan setting direktori kerja melalui **Project Properties**. Pilih **Run** pada pilihan kotak **Categories**. Tentukan direktori kerja dengan menekan tombol **Browse...** pada kotak masukan **Working Directory**.

Gambar 1.32 Penentuan Direktori Kerja

9. Lakukan proses *Clean and Build*, dan perhatikan hasilnya.